[TS] TypeScript Playground 만들기

by Jbee

최소한의 npm과 설정으로 TypeScript playground 만들기를 진행해보겠습니다. 사실 가장 빠른 playground 세팅은 이 저장소를 clone하면 됩니다! ㅎㅎ

- https://github.com/JaeYeopHan/typescript_playground

우선적으로 `TypeScript` 님부터 모셔오도록 합시다 :)

```
$ npm install -g typescript
```

프로젝트 세팅

디렉토리를 하나 만들고 `npm init`을 통해 `package.json` 설정을 해줍니다.

```
$ mkdir typescript_playground && cd typescript_playground
$ npm init
# set configuration of npm
```

```
"start": "webpack-dev-server --open"
```

transpiler를 태울 webpack이 필요한데 그 config파일과 TypeScript 설정 파일인 `tsconfig.json` 파일을 생성해줍니다.

```
$ touch webpack.config.js tsconfig.json
$ touch index.html
$ mkdir src
$ touch src/index.ts
```

다음은 개인 취향에 따른 optional한 세팅 파일들입니다.

```
# And create optional files...
$ touch .editorconfig
$ touch .gitignore
$ touch README.md
```

이 파일들의 내용은 repo를 참고하세요!

이젠 필요한 npm 모듈들을 설치합니다.

```
$ npm install webpack webpack-dev-server --save-dev
$ npm install awesome-typescript-loader --save-dev
```

프론트엔드 개발자의 영원한 친구 `we1bpack`과 그 친구 `webpack-dev-server`를 설치해줍니다. 그리고 공식 문서에는 `ts-loader`를 사용하던데요, 저희는 가볍게 무시하고 `awesome-typescript-loader`를 설치해줍니다. 딱 이만큼만 설치하면 됩니다. ES6를 한 번 사용해보려고 `babel`이며 `babel-core`며 `.babelrc`파일이며 온갖 이상한 npm파일과 설정 파일을 생성한 기억이 한 번쯤은 있으실텐데요, 타입스크립트는 여기까지가 끝입니다. (행복합니다.)

설정 파일 작성하기

이제 각종 config 파일들을 작성해봅시다. 물론 설정 파일 또한 최소한으로 작성합니다.

webpack.config.js

```
const webpack = require('webpack');
const path = require("path");
const { CheckerPlugin } = require('awesome-typescript-loa
```

webpack.config.js

```
module.exports = {
    entry: './src/index.ts',
    output: {
        filename: './dist/bundle.js'
    },
    resolve: {
        extensions: ['.ts', '.tsx', '.js', '.jsx']
    },
    devtool: 'inline-source-map',
    module: {
        loaders: [
            {
                test: /\.ts?$/,
                loader: 'awesome-typescript-loader',
                include: path.resolve(__dirname, 'src')
            }
        ]
    },
    plugins: [
        new CheckerPlugin(),
        new webpack.HotModuleReplacementPlugin(),
    ],
};
```

이 문서에서는 webpack tutorial을 다루지 않습니다. webpack tutorial은 여기에서 따라하실 수 있습니다. (깨알홍보!)
`HotMoudleReplacementPlugin` 이 녀석은 도저히 포기할 수 없어서 추가했습니다. 이제 본격적으로 typescript 설정 들어갑니다.

- https://github.com/JaeYeopHan/webpack2_tutorial

tsconfig.json

```
{
    "compilerOptions": {
        "noImplicitAny": true,
        "removeComments": true
    },
    "awesomeTypescriptLoaderOptions": {
        /* ... */
    }
}
```

ES6를 처음 시작하던 때의 `.babelrc` 파일이 기억나시나요? 그것과 비슷하다고 볼 수 있는데요, 상당히 간단합니다. 거의 모든 부분을 `awesome-typescript-loader` 에서 cover하고 있고 입맛에 맞게 customize할 수 있도록 `"awesomeTypescriptLoaderOptions"` 라는 이름으로 인터페이스가 열려있습니다.

index.html

javascript로 transpile될 typescript파일을 브라우저 개발자 도구에서 확인하기 위해서 `index.html` 파일을 간단히 작성해줍니다.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
</head>
<body>
    Hello, TypeScript world!!
    <script src="/dist/bundle.js"></script>
</body>
</html>
```

왠만한 설정은 모두 끝났습니다. webpack에 설정한 loader를 잘 타는지 확인하기 위해 index.ts에 console한 번 찍어봅니다.

```
console.log(`hello typescript world`);
```

그리고 start!

```
$ npm start
```

브라우저가 자동으로 실행될텐데요, 개발자 도구를 열어보면 hello typescript world라는 console이 찍혀있을 겁니다. 이제 TypeScript로 해볼 수 있는 것을 이것 저것 해보면 됩니다.

감사합니다.

Reference

- GitHub of awesome-typescript-loader
- GitHub of tslint-loader
- Jbee's Webpack tutorial
- TypeScript Official Document
- GitHub of jest